容器监控及优化

胡朝旭

美团点评 | 技术团队

内容提要

容器概要

性能监控

体验优化

容器概要

Hybrid诞生背景

A 开发成本

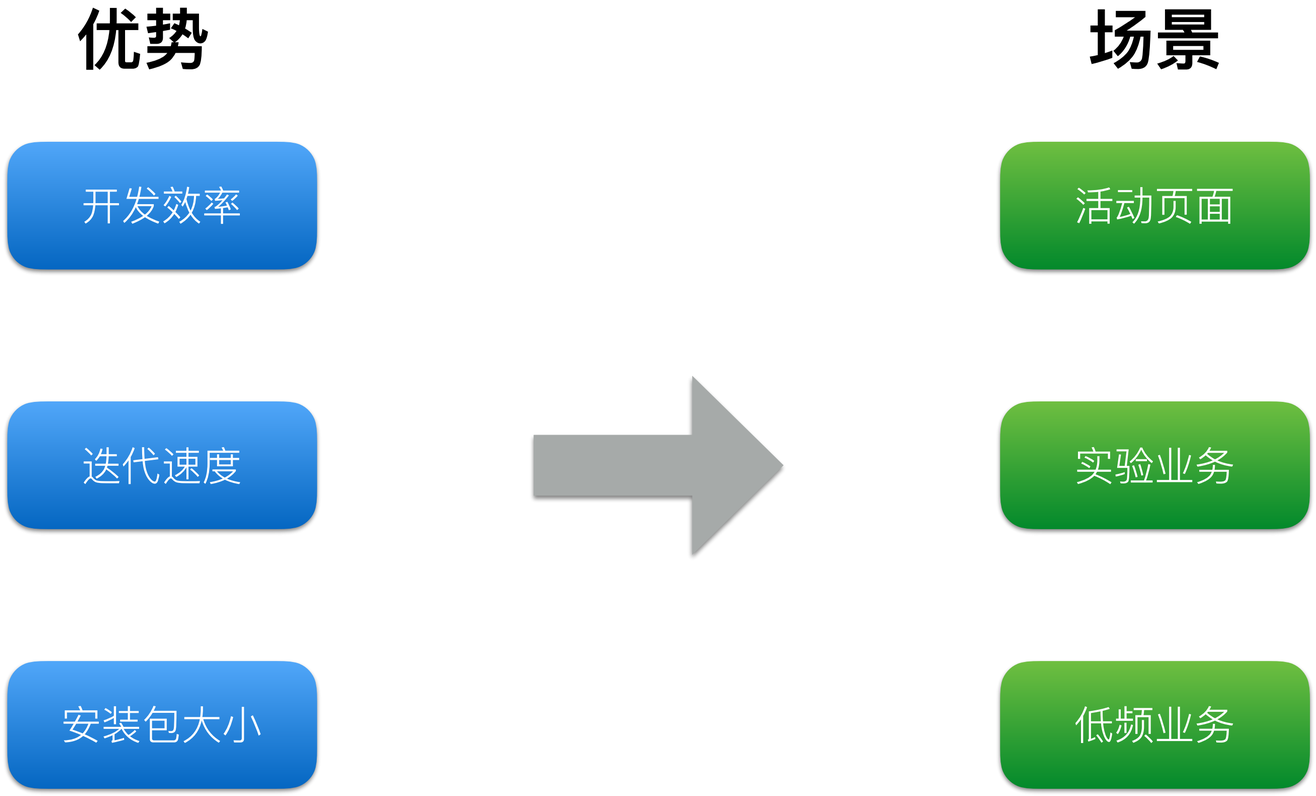
使用Native开发，需要兼顾Android、iOS、PC端及其它操作系统，导致开发成本高企。

B 迭代速度

功能变更、故障修复强烈依赖发版。新特性不能及时同步到用户。

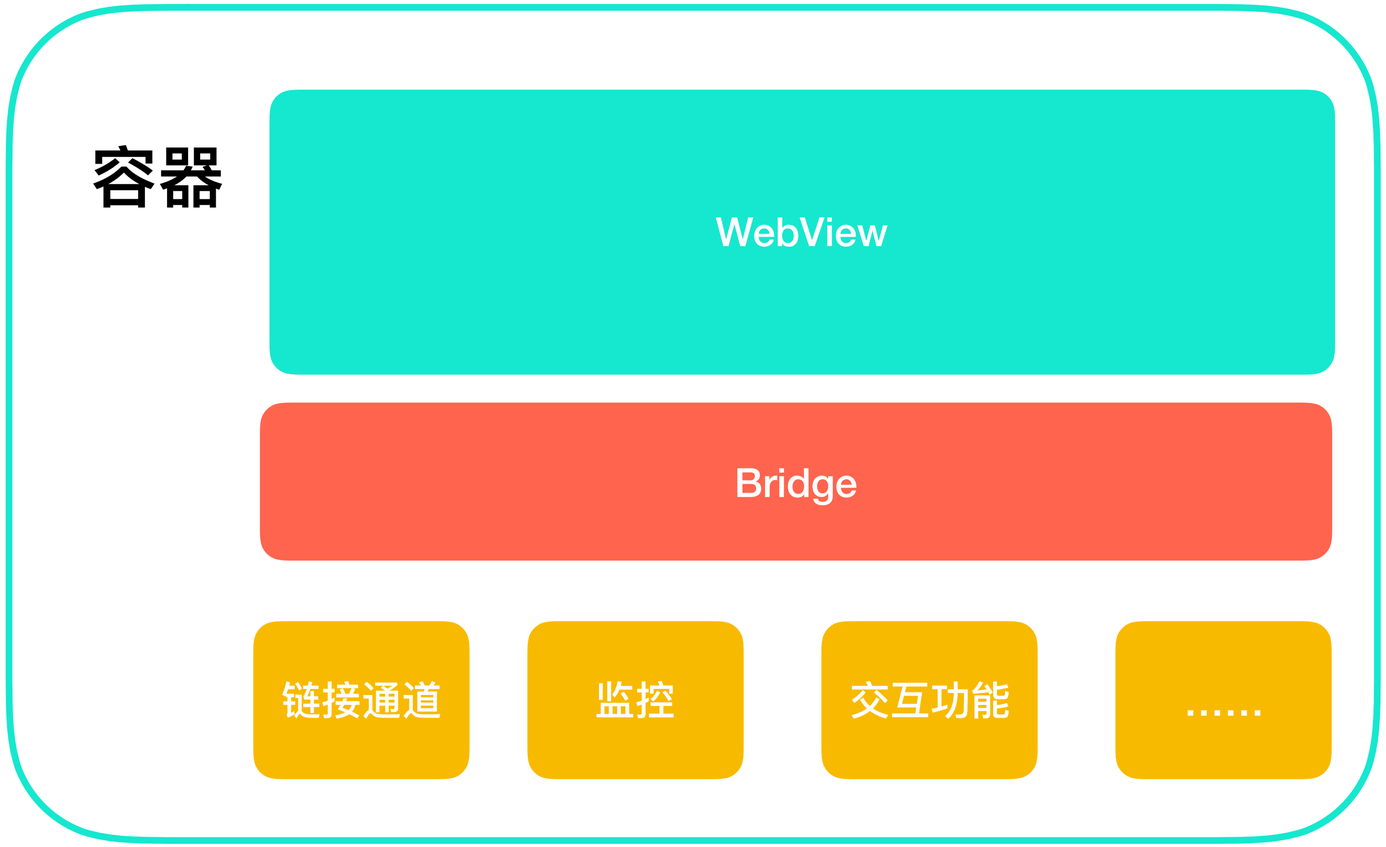
容器概要

Hybrid的定位



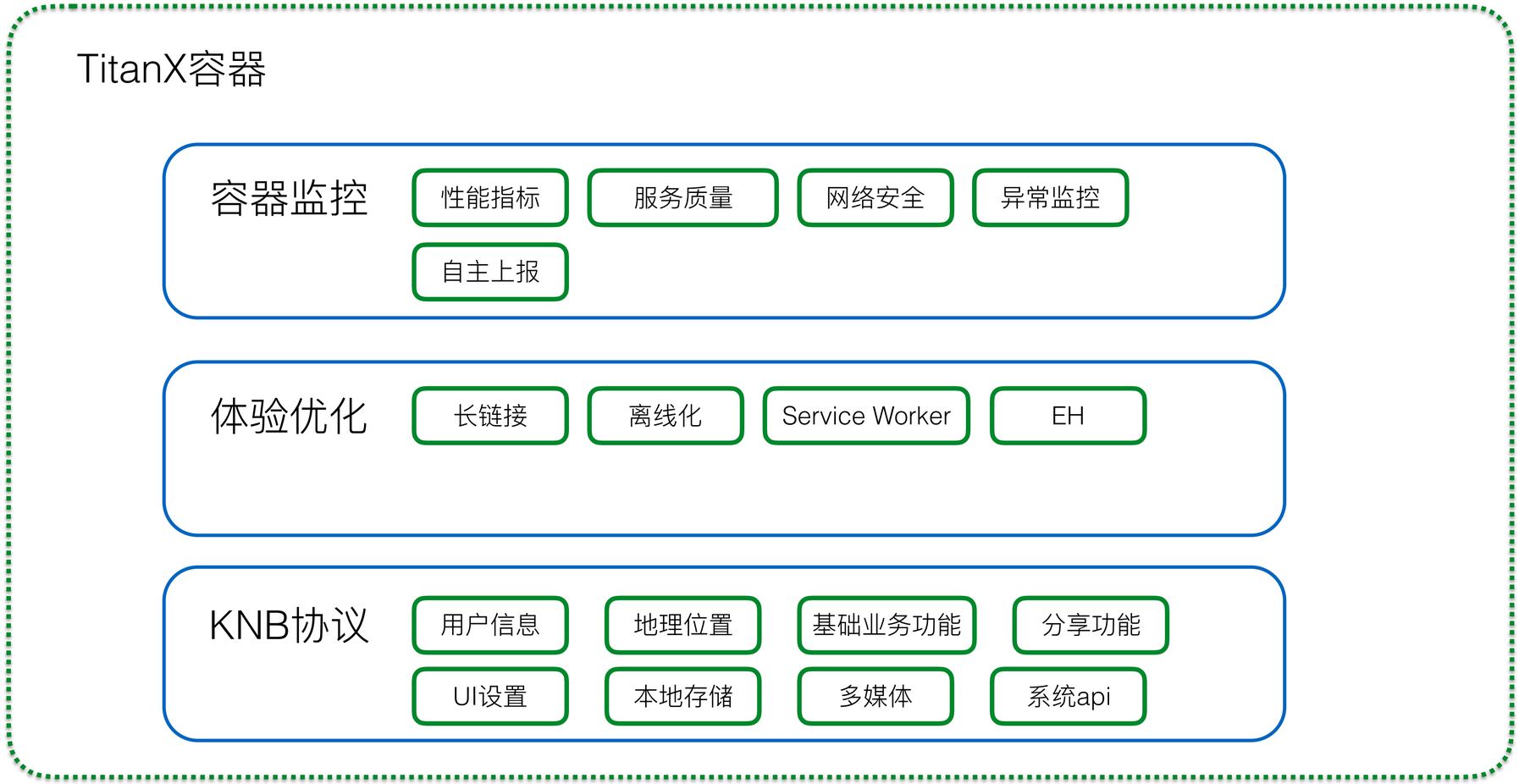
容器概要

容器的定义



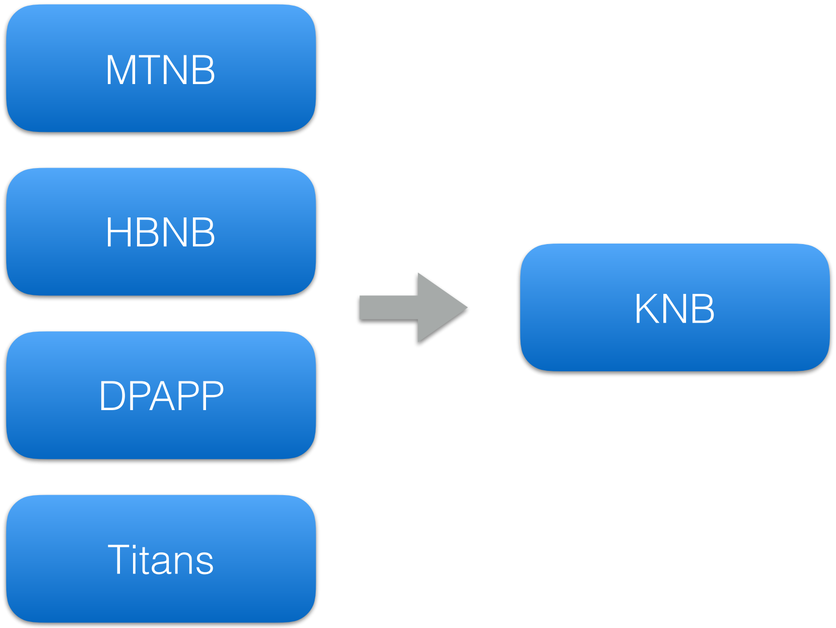
容器概要

TitanX容器



容器概要

KNB协议



在页面侧提供统一的api

在原生侧提供差异实现

内容提要

容器概要

性能监控

体验优化

容器监控

内容概要

01 监控指标

一个Hybrid页面的体验、性能受很多因素影响，选取合理的指标对后续的优化方向格外重要。主要会介绍监控的数据和我们使用数据的方式

02 监控案例

介绍一些新美大对容器内监控的内容和实现方式

03 触发时机

部分监控对于用户性能会有较大开销，选择合适的时机尤为重要。这里介绍量个我们触发的时机

容器监控

影响因素

不可控部分

- OS
- 运营商
- 地域
- 机型

用户体验

可控部分

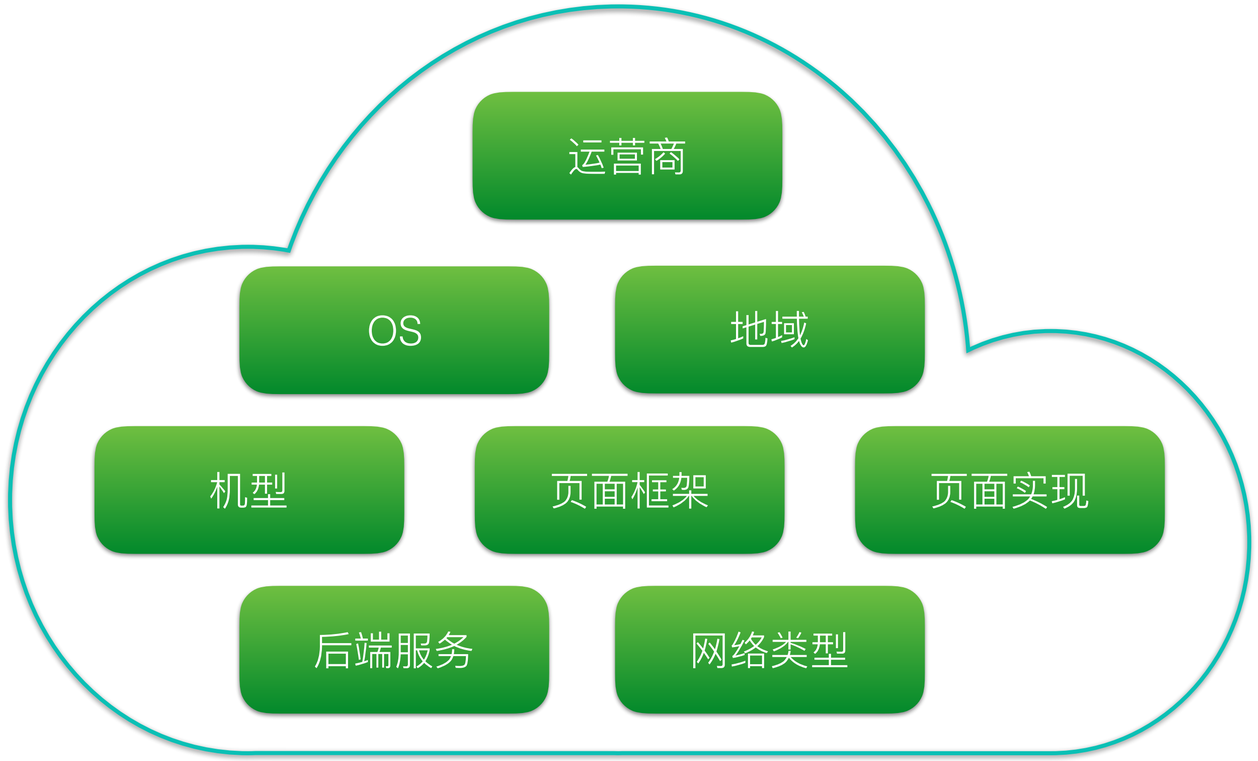
- 页面框架
- 页面实现
- 网络类型
- 后端服务

选取哪类监控指标检测影响因素

以何种形式利用这些监控指标

容器监控

页面性能维度



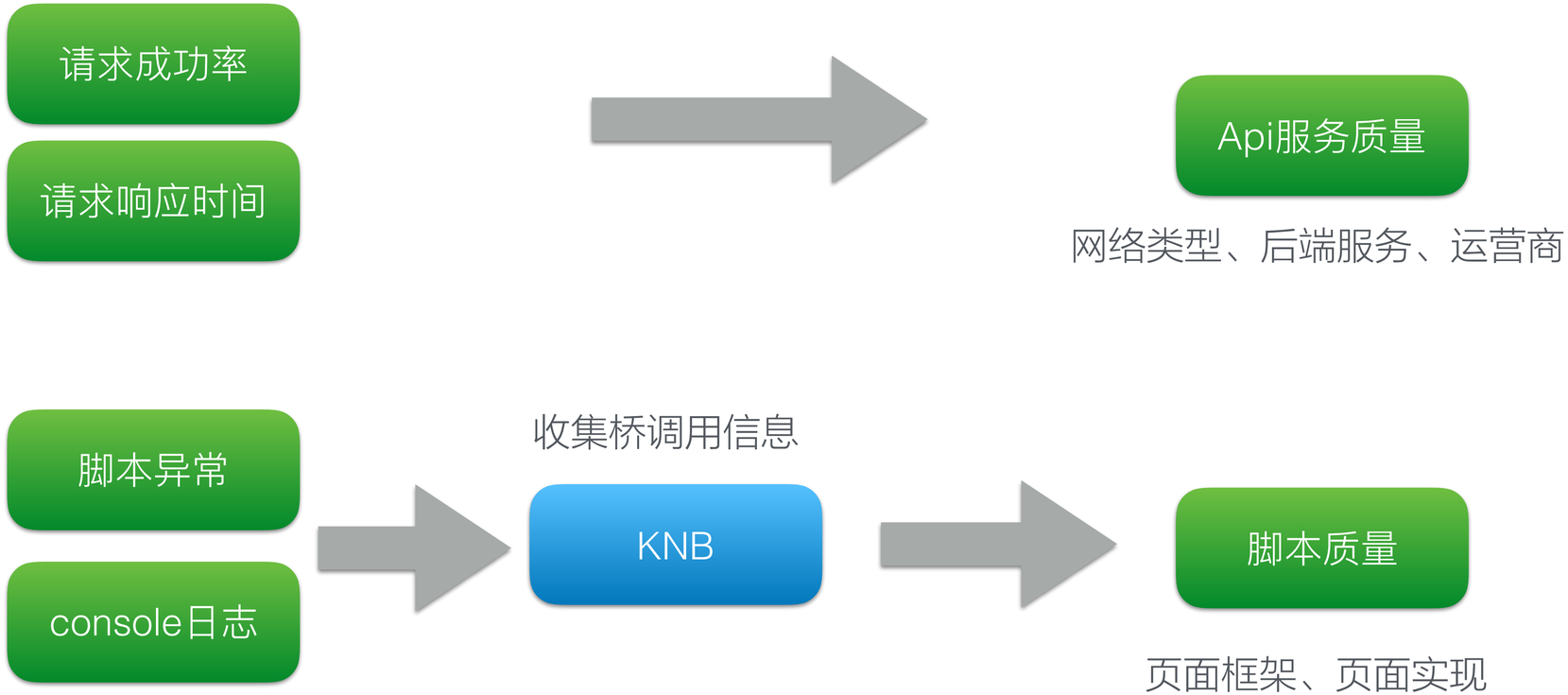
90分位数

前端上报

- 首字节时间
- 可交互时间
- 完全加载时间
- Ajax时间

页面监控

服务质量维度


请求成功率
请求响应时间
Api服务质量
网络类型、后端服务、运营商
脚本异常
console日志
收集桥调用信息
KNB
脚本质量
页面框架、页面实现

页面监控

网络安全维度

面临的问题

- 内容篡改
- 页面重定向
- 请求拦截
- DNS劫持
- 用户数据抓取
- ……

常用的方案

- https
- 长链接
- IP替代域名
- CSP
- 资源离线化
- Traceroute追踪

常用方案都伴随着对应的代价

页面监控

触发时机I-页面完整性

注入脚本都对分块加载带来影响

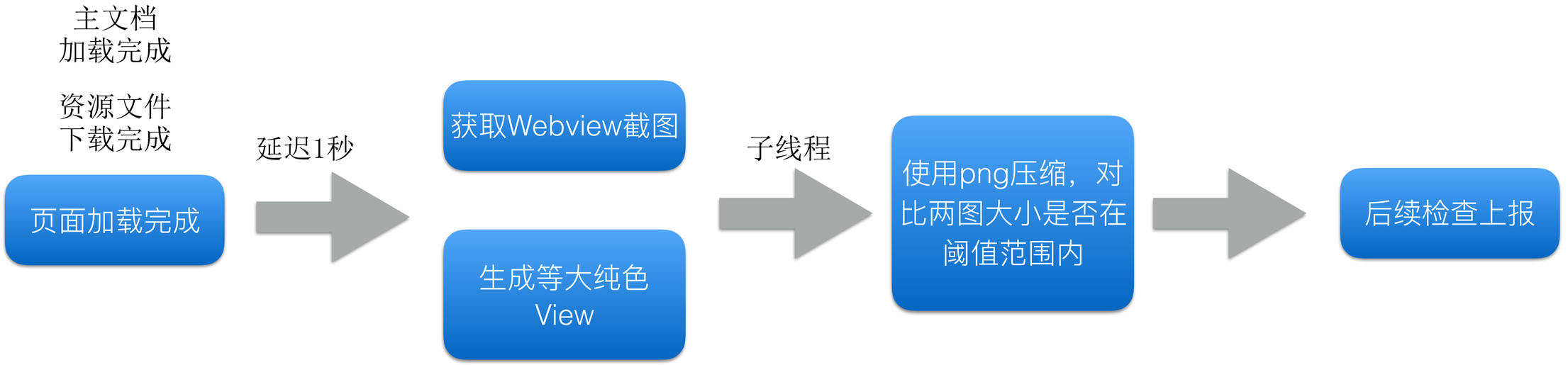
页面监控

触发时机II-白屏监控

- 白屏监控场景
  - 不会触发前端异常
  - 不触发webview异常回调
  - 用户体验最差

页面监控

触发时机II-白屏监控



内容提要

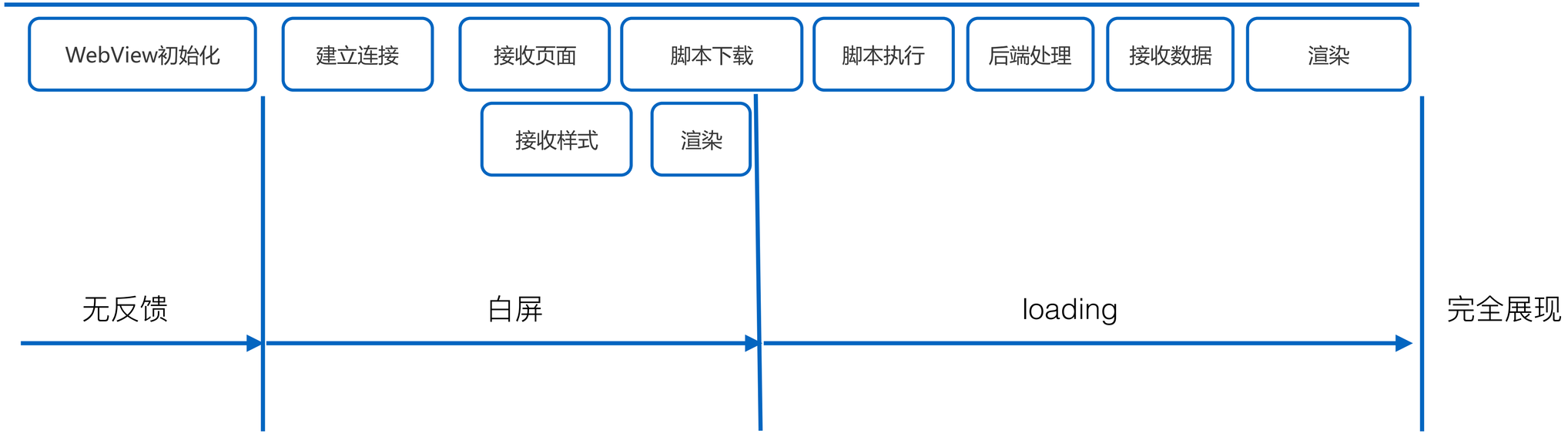
容器概要

性能监控

体验优化

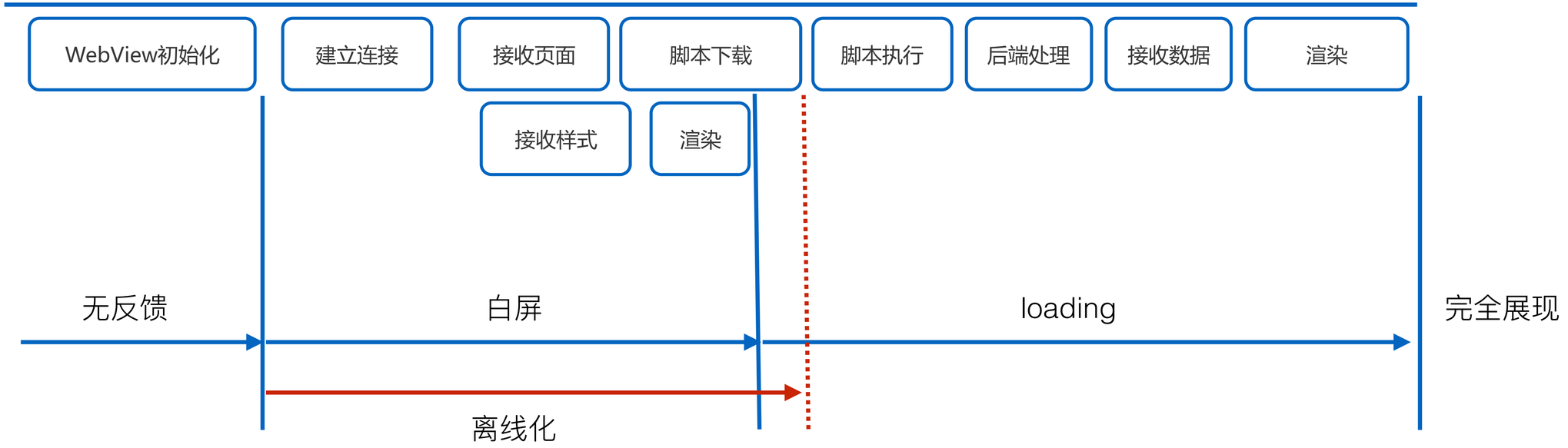
体验优化

页面加载流程


WebView初始化
建立连接
接收页面
脚本下载
脚本执行
后端处理
接收数据
渲染
接收样式
渲染
无反馈
白屏
loading
完全展现

体验优化

离线化


WebView初始化
建立连接
接收页面
脚本下载
脚本执行
后端处理
接收数据
渲染
接收样式
渲染
无反馈
白屏
loading
完全展现
离线化

体验优化

离线化

首要收益

加载速度

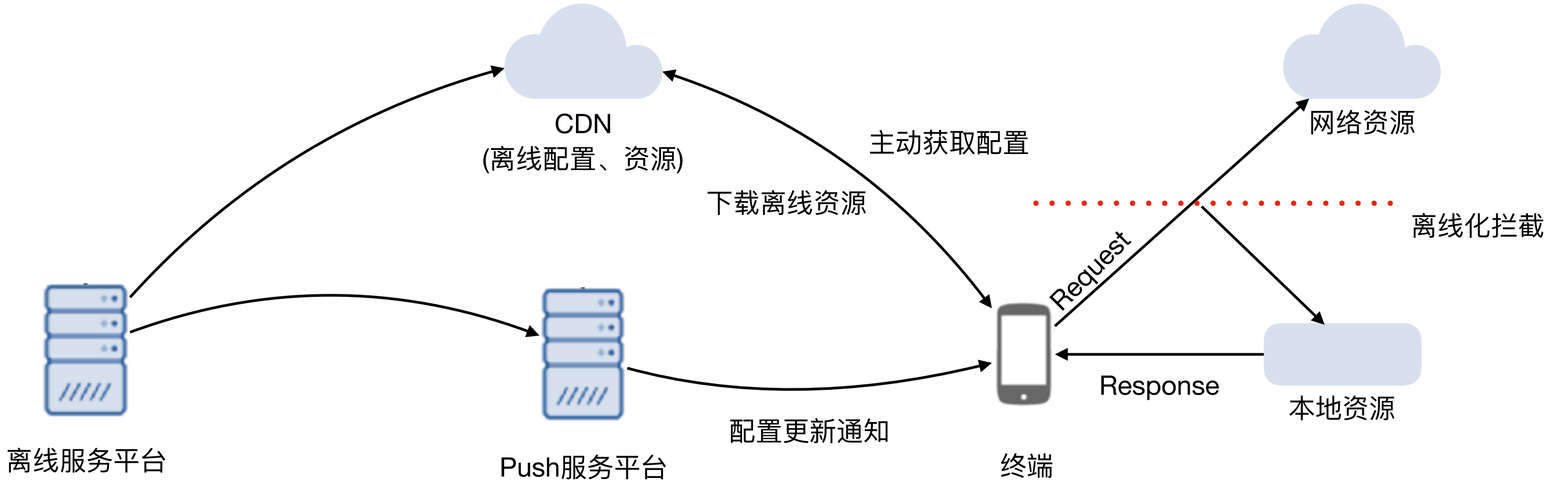
附加收益

资源正确性

流量收益

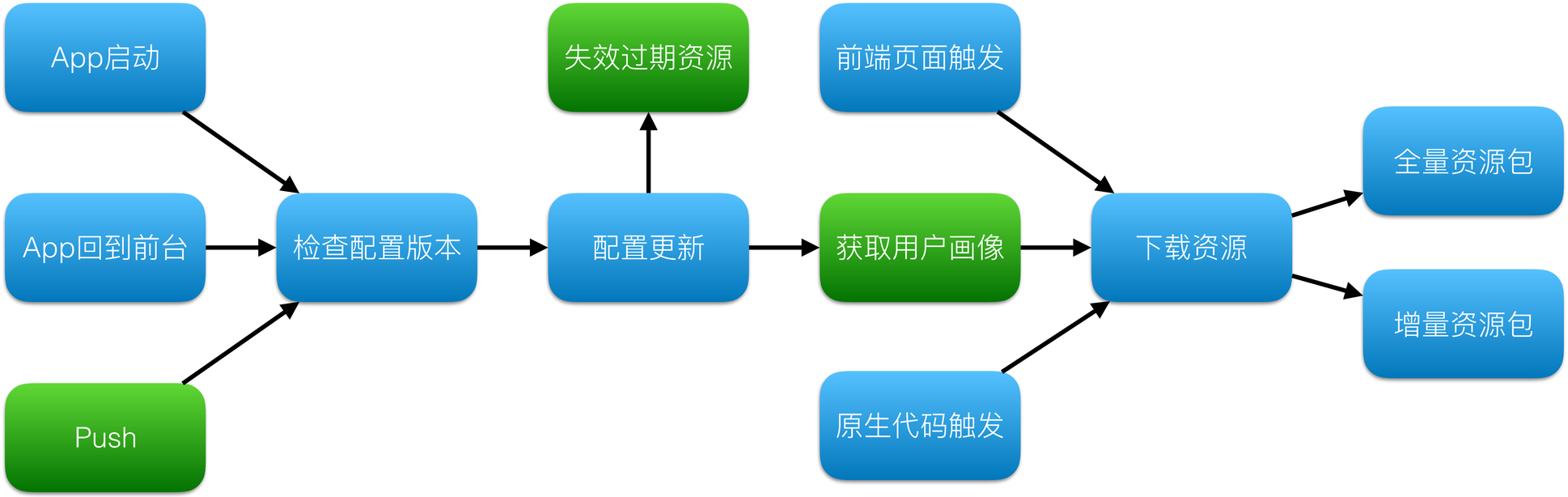
体验优化

资源离线化—系统架构



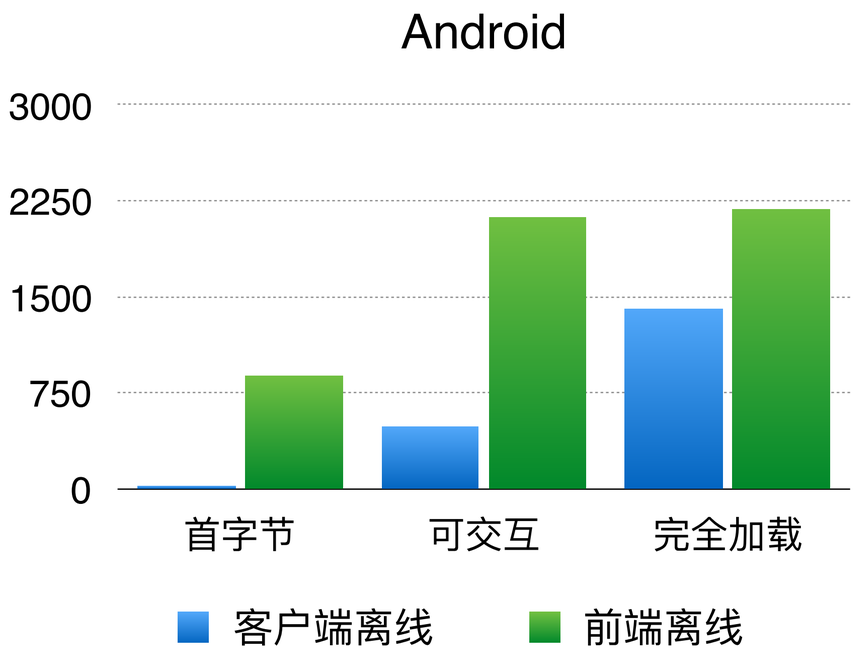
体验优化

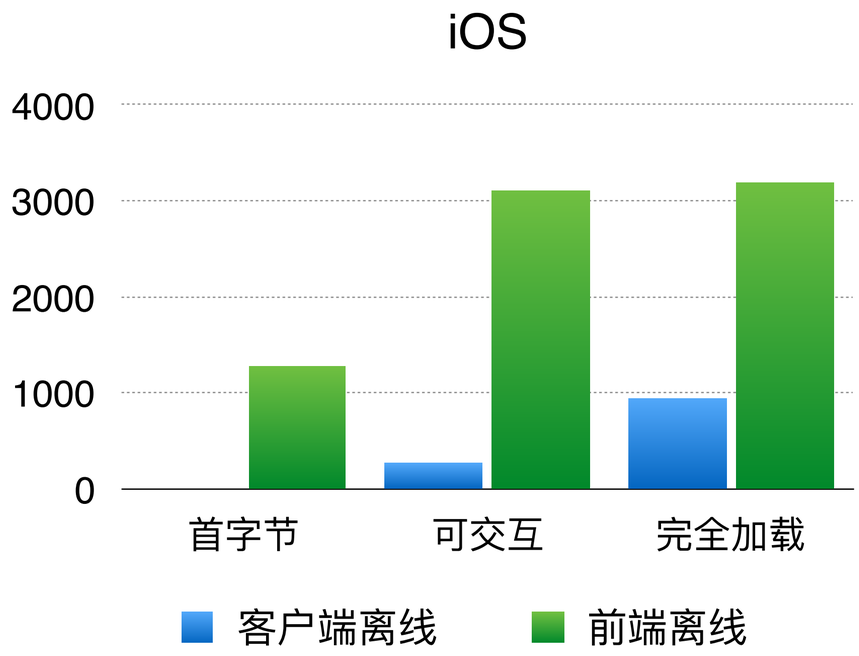
资源离线化—体验优化



体验优化

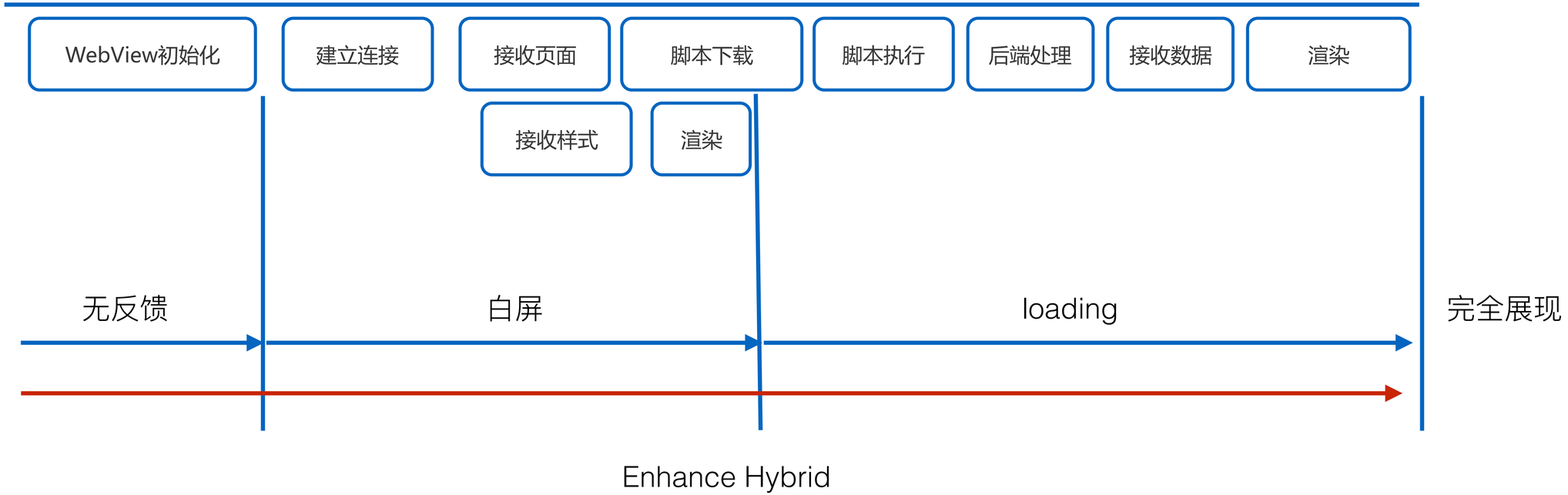
资源离线化—首次加载收益





体验优化

Enhance Hybrid



体验优化

Enhance Hybrid

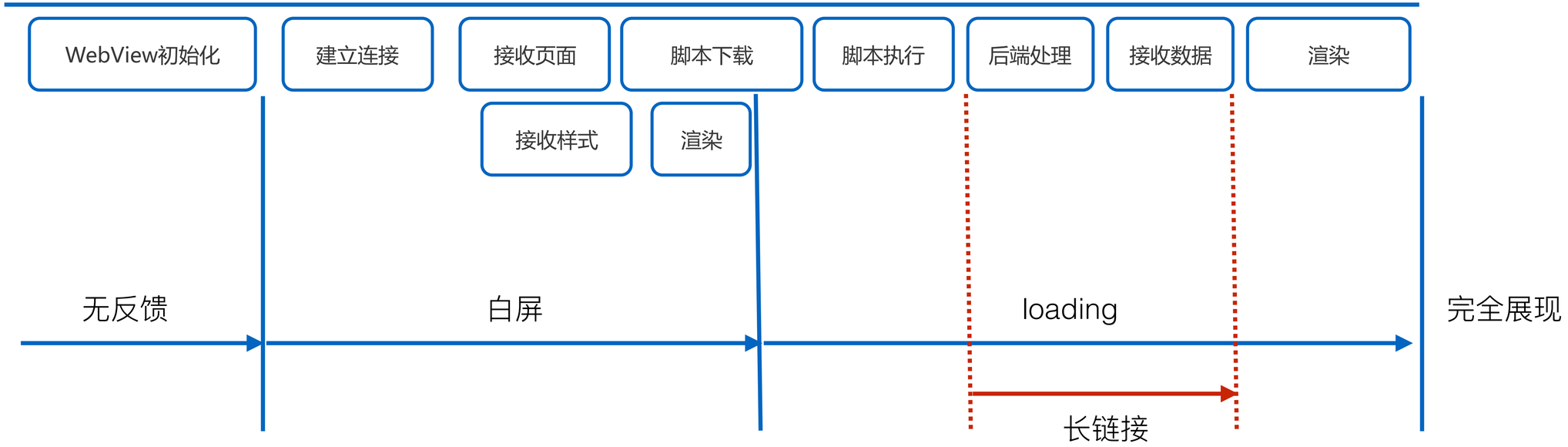
体验优化

Enhance Hybrid

让前端Web页面有近似Native的体验

体验优化

长链接


WebView初始化
建立连接
接收页面
脚本下载
脚本执行
后端处理
接收数据
渲染
接收样式
渲染
无反馈
白屏
loading
完全展现
长链接

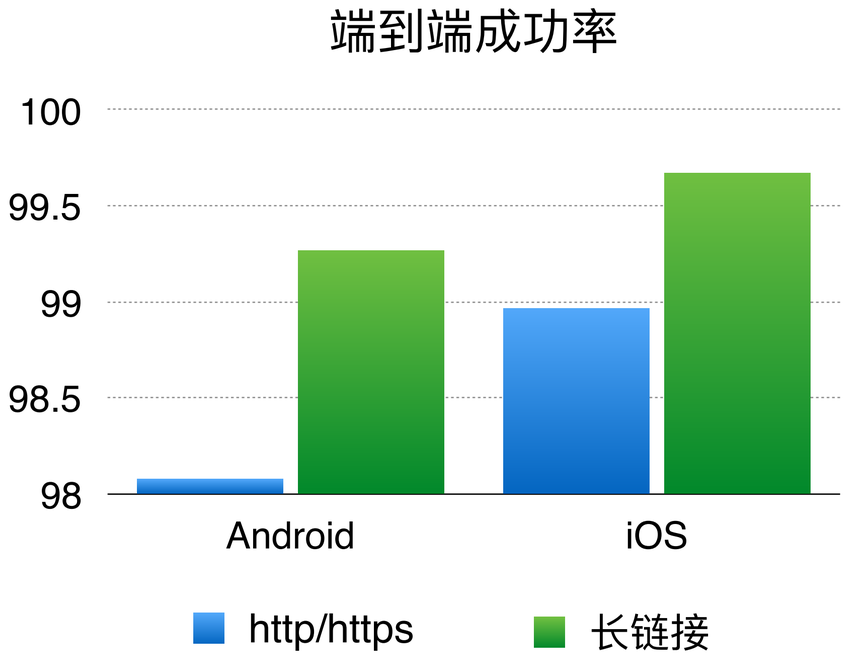
体验优化

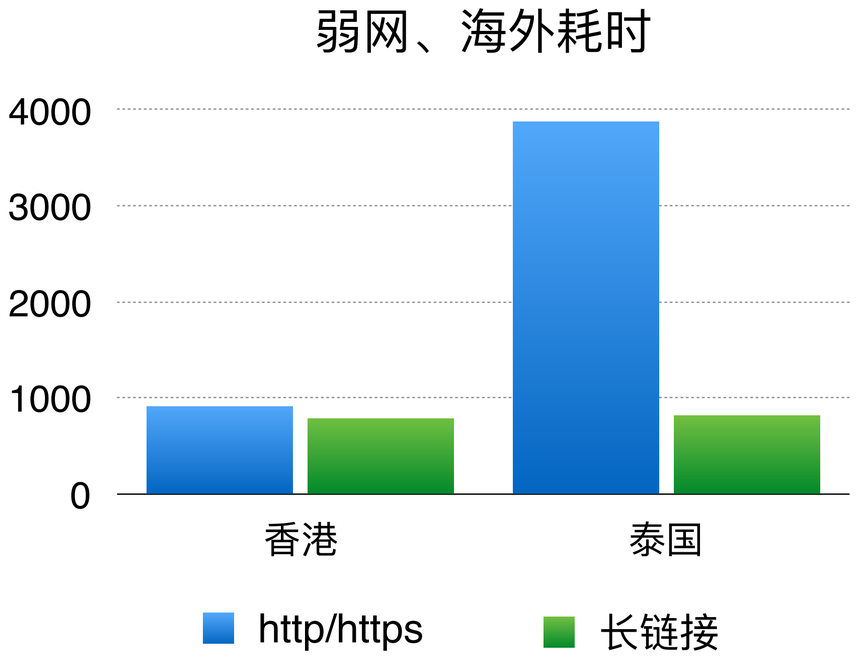
长链接

- 应用场景
  - 服务接口访问
- 优化点
  - 节省链接建立开销
  - 专线传输保证海外、弱网场景
  - 提升传输效率
- 数据安全性
  - 防止DNS劫持
  - 防止传输数据被篡改、监控
  - 提升数据抓取复杂度

体验优化

长链接—端到端提升





体验优化

安全优化

	https	离线化	长链接
DNS劫持	不支持	支持	支持
页面重定向	支持	支持	支持
防篡改	支持	支持	支持
防拦截	不支持	支持	支持
其它	无	不支持动态数据	防数据抓取，不支持CDN加速

谢谢大家

Q&A